Serial No.: 09/355,214
Filed: July 23, 1999.

requested.

Attached hereto is an appendix entitled "Version with Markings to Show Changes Made" which depicts the changes made to the instant application by the current amendment.

Objections

Applicants acknowledge the Examiner's remarks concerning the Declaration. Accompanying this Amendment and Response is a new Declaration executed by inventor Andrew Chan, which correctly identifies U.S. Patent Application Serial No. 08/819,013 as patented and U.S. Patent Application Serial No. 08/788,322 as abandoned. We are presently awaiting a newly executed declaration from the second inventor, Chong Fu.

Applicants acknowledge the Examiner's objection to the incorporation by reference of subject matter deemed essential. The subject matter in question concerns the nature of high stringency conditions for hybridization, which is deemed essential because "high stringency conditions" is recited in the pending claims. Applicants point out that the pending claims have been cancelled, and that the new amended claims do not recite for BLNK proteins encoded by nucleic acids that will hybridize under high stringency conditions to other nucleic acids.

Applicants request withdrawal of the objections.

Rejections Under 35 U.S.C. §101

Claims 23-34 stand rejected under 35 U.S.C. §101 as lacking either a specific and substantial asserted utility or a well-established utility. The Examiner expresses that BLNK protein activity is not defined by the instant specification and that the utility asserted for BLNK protein by the instant specification is a general utility. Applicants traverse.

Applicants draw the Examiner's attention to the revised U.S. PTO Utility Examination Guidelines published in the Federal Register, vol. 66, No. 4. At page 1098 of the identified volume of the Federal Register, at section B, 2, (2), the guidelines state:

Serial No.: 09/355,214
Filed: July 23, 1999.

> An applicant need only provide one credible assertion of specific and substantial utility for each claimed invention to satisfy the utility requirement.

Further, at page 1096 in response to comment 19, the Commissioner cites *Fujikawa v. Wattanasin*, 95 F. 3d 1559,1562, 39 USPQ2d 1895, 1900 (Fed. Cir. 1996), which states:

> "[A] 'rigorous correlation' need not be shown in order to establish a practical utility; 'reasonable correlation' is sufficient."

Further, at page 1098, section B, 4, the guidelines state:

> Office personnel are reminded that they must treat as true a statement of fact made by an applicant in relation to an asserted utility, unless countervailing evidence can be proved that shows that one of ordinary skill in the art would have a legitimate basis to doubt the credibility of such a statement. Similarly, Office personnel must accept an opinion from a qualified expert that is based upon relevant facts whose accuracy is not being questioned: it is improper to disregard the opinion solely because of a disagreement over the significance or meaning of the facts offered.

The instant specification asserts a number of characteristics and functions for BLNK proteins that support that the claimed BLNK protein compositions have specific, substantial utility. For example, the instant specification asserts at page 6, lines 11-15 that BLNK protein is tyrosine phosphorylated by Syk following B cell receptor activation, and at page 20, lines 6-8 that BLNK protein binds to Grb2, PLCγ, Nck and Vav, and regulates calcium levels and modulates cytoskeletal organization, and at page 19, lines 28-29 that BLNK protein is critical for B cell receptor mediated response and B cell function. Applicants submit that these statements are credible and should be accepted.

The instant application also provides methods for using the claimed BLNK protein compositions, for example to screen for bioactive agents that are capable of modulating BLNK protein activity (page 23, lines 4-11).

Applicants submit that the asserted function of BLNK protein is specific, as the asserted binding activities and B cell regulation activities disclosed in the instant application are not properties shared by all proteins.

In addition, Applicants submit that the asserted utility of BLNK protein is

Serial No.: 09/355,214
Filed: July 23, 1999.

substantial, as the ability to modulate B cell function and to identify bioactive agents therefore is clearly desirable.

While Applicant submits the assertions of the specification should be accepted without any further discussion into their accuracy, Applicant further submits support of the accuracy of the assertions in the form of the enclosed Declaration under 1.132 (the declaration). In paragraph 5 of the declaration, the inventor Andrew Chan, Ph.D., representing one of ordinary skill in the art, declares that he would expect to be able to use the claimed BLNK protein compositions as provided for in the present application. Moreover, the declaration discusses data already of record which confirms the accuracy of the assertions made in the application. Specifically, the declaration shows that loss of BLNK gene function results in abnormal B cell function, supporting the assertion that BLNK protein is a modulator of B cell function and that the loss thereof results in a BLNK-mediated disorder, and that the claimed BLNK protein compositions have specific and substantial utility.

Claims 23-34 have been cancelled without prejudice, disclaimer or admission. New Claims 35-38 are directed to BLNK proteins comprising an amino acid sequence having at least about 95% identity to SEQ ID NO:1. Claims 36-38 are further directed to BLNK proteins comprising SEQ ID NO:1, BLNK proteins which will bind to specified BLNK protein binding partners, and BLNK proteins which lack specific tyrosine phosphorylation sites as set forth in SEQ ID NO:1, respectively.

Claims 39-41 are directed to BLNK proteins comprising an amino acid sequence having at least about 95% identity to the amino acid sequence encoded by SEQ ID NO:2. Claims 40 and 41 are further directed to BLNK proteins comprising an amino acid sequence encoded by SEQ ID NO:2, and BLNK proteins which will bind to specified BLNK protein binding partners, respectively.

Claim 42 is directed to a pharmaceutical composition comprising the BLNK protein according to any one of Claims 35-41.

Claim 43 is directed to an antibody that binds to the BLNK protein according to any one of Claims 35-41.

Claims 44 and 45 are directed to methods for using BLNK proteins which BLNK

Serial No.: 09/355,214
Filed: July 23, 1999.

proteins comprise an amino acid sequence having at least about 95% identity to the amino acid sequence set forth in SEQ ID NO:1 and will bind to Grb2, PLCγ, Vav, or Nck. Claim 44 is directed to methods for screening for a bioactive agent that will bind to a BLNK protein, while Claim 45 is directed to methods for screening for a bioactive agent capable of modulating BLNK protein activity.

Applicants submit that the new claims satisfy the utility requirement of 35 U.S.C. §101 and request withdrawal of the rejection and allowance of the claims.

Rejections Under 35 U.S.C. §112, First Paragraph - How to Use

Claims 23-34 stand rejected under 35 U.S.C. §112, first paragraph as failing to teach the reasonably skilled artisan how to use the invention for a credible, specific and substantial utility. Applicants traverse.

As discussed above and supported by the accompanying declaration, Applicants submit that new Claims 35-45 satisfy the utility requirement of 35 U.S.C. §101. Accordingly, Applicants submit that a person of reasonable skill in the art would be able to use the invention in full scope of the claims for a credible, specific and substantial utility.

Applicants request withdrawal of the rejection and allowance of the new claims.

Rejections Under 35 U.S.C. §112, First Paragraph - Written Description

Claims 23-34 stand rejected under 35 U.S.C. §112, first paragraph as lacking written description support in the specification. Applicants traverse.

The Office Action expresses that Claims 23-34 are directed to a very large genus of recombinant polypeptide species, uses thereof, and antibodies that bind thereto.

Claims 23-34 have been cancelled without prejudice, disclaimer or admission.

Applicants have amended the claims to further define the scope of the claimed BLNK protein compositions. Claims 35-38 are directed to BLNK proteins comprising an amino acid sequence having at least about 95% identity to SEQ ID NO:1. Claims 36-38 are further directed to BLNK proteins comprising SEQ ID NO:1, BLNK proteins which will bind to specified BLNK protein binding partners, and BLNK proteins which lack

Serial No.: 09/355,214
Filed: July 23, 1999.

specific tyrosine phosphorylation sites as set forth in SEQ ID NO:1, respectively.

Claims 39-41 are directed to BLNK proteins comprising an amino acid sequence having at least about 95% identity to the amino acid sequence encoded by SEQ ID NO:2. Claims 40 and 41 are further directed to BLNK proteins comprising an amino acid sequence encoded by SEQ ID NO:2, and BLNK proteins which will bind to specified BLNK protein binding partners, respectively.

Claim 42 is directed to a pharmaceutical composition comprising the BLNK protein according to any one of Claims 35-41.

Claim 43 is directed to an antibody that binds to the BLNK protein according to any one of Claims 35-41.

Claims 44 and 45 are directed to methods for using BLNK proteins which BLNK proteins comprise an amino acid sequence having at least about 95% identity to the amino acid sequence set forth in SEQ ID NO:1 and will bind to Grb2, PLCγ, Vav, or Nck. Claim 44 is directed to methods for screening for a bioactive agent that will bind to a BLNK protein, while Claim 45 is directed to methods for screening for a bioactive agent capable of modulating BLNK protein activity.

The Office Action expresses at page 5 that the instant specification does not provide sufficient teaching or guidance for one of reasonable skill in the art to determine sequences that are within the scope of 95% identity to SEQ ID NO:1 and SEQ ID NO:2. The Examiner asserts that because no specific algorithm is disclosed, the claims do not find written description support in the specification. Applicants disagree.

Applicants point out that specific algorithms are disclosed in the instant specification. At page 24, lines 6-7, the instant application states: "All references cited herein are incorporated by reference."

Further, at page 5, line 12, the specification cites the prior art of Altschul et. al., J. Mol. Biol. 215:403-410, 1990 (Altschul-A) (a copy of which is attached as Exhibit A). Altschul-A describes the basic local alignment search tool (BLAST) and at page 404, left column, paragraph 3 discloses parameters for measuring nucleic acid similarity.

Altschul-A also discloses the prior art of Altschul et. al., J. Mol. Biol. 219:555-565, 1991 (Altschul-B) (a copy of which is attached as Exhibit B). Altschul-B discloses

Serial No.: 09/355,214
Filed: July 23, 1999.

the PAM-120 amino acid substitution matrix and scoring parameters therefore. The PAM-120 matrix and parameters are found at page 560, Table 4.

Accordingly, Applicants submit that the present application does properly disclose sequence comparison algorithms.

Applicants submit that one of ordinary skill in the art would clearly construe the meaning of 95% sequence identity based on the teaching of the specification and the knowledge held in the art, and would conclude that Applicants were in possession of the claimed BLNK protein compositions at the time of filing of the priority application.

Applicants submit that Claims 35-45 satisfy the written description requirement of 35 U.S.C. §112, first paragraph and request withdrawal of the rejection and allowance of the claims.

Rejections Under 35 U.S.C. §112, Second Paragraph - Indefiniteness

Claims 1-22, 23, 25, 27-28, and 31-34 stand rejected under 35 U.S.C. §112, second paragraph as being indefinite. In particular, Claims 23, 25, 27-28 and 31-34, under consideration in the case, are found indefinite for use of the following phrases:

i) "polypeptide comprising the protein" (Claim 23);

ii) "high stringency conditions" (Claim 25 and 28); and

iii) "polypeptide" (Claims 27, 31-33).

As a preliminary matter, Applicants point out that Claims 23-34 have been cancelled without prejudice, disclaimer or admission.

New Claims 35-44 do not recite for proteins encoded by nucleic acids that will hybridize under high stringency conditions.

Applicants request withdrawal of the rejection and allowance of the new claims.

CONCLUSION

Applicants submit that the application is now in form for allowance and early notification of such is requested. If there remain issues that the Examiner believes may be resolved by telephone, he/she is respectfully requested to contact the undersigned at (415)

Serial No.: 09/355,214
Filed: July 23, 1999.

781-1989.

Respectfully submitted,

FLEHR HOHBACH TEST
ALBRITTON & HERBERT LLP

Richard F. Trecartin
Reg. No. 31,801

Four Embarcadero Center, Suite 3400
San Francisco, CA 94111-4187
Telephone: (415) 781-1989

Dated: Nov 8, 2001

1064013

J. Mol. Biol. (1990) 215, 403-410

Basic Local Alignment Search Tool

Stephen F. Altschul[1], Warren Gish[1], Webb Miller[2] Eugene W. Myers[3] and David J. Lipman[1]

[1]*National Center for Biotechnology Information National Library of Medicine, National Institutes of Health Bethesda, MD 20894, U.S.A.*

[2]*Department of Computer Science The Pennsylvania State University, University Park, PA 16802, U.S.A.*

[3]*Department of Computer Science University of Arizona, Tucson, AZ 85721, U.S.A.*

(Received 26 February 1990; accepted 15 May 1990)

A new approach to rapid sequence comparison, basic local alignment search tool (BLAST), directly approximates alignments that optimize a measure of local similarity, the maximal segment pair (MSP) score. Recent mathematical results on the stochastic properties of MSP scores allow an analysis of the performance of this method as well as the statistical significance of alignments it generates. The basic algorithm is simple and robust; it can be implemented in a number of ways and applied in a variety of contexts including straightforward DNA and protein sequence database searches, motif searches, gene identification searches, and in the analysis of multiple regions of similarity in long DNA sequences. In addition to its flexibility and tractability to mathematical analysis, BLAST is an order of magnitude faster than existing sequence comparison tools of comparable sensitivity.

1. Introduction

The discovery of sequence homology to a known protein or family of proteins often provides the first clues about the function of a newly sequenced gene. As the DNA and amino acid sequence databases continue to grow in size they become increasingly useful in the analysis of newly sequenced genes and proteins because of the greater chance of finding such homologies. There are a number of software tools for searching sequence databases but all use some measure of similarity between sequences to distinguish biologically significant relationships from chance similarities. Perhaps the best studied measures are those used in conjunction with variations of the dynamic programming algorithm (Needleman & Wunsch, 1970; Sellers, 1974; Sankoff & Kruskal, 1983; Waterman, 1984). These methods assign scores to insertions, deletions and replacements, and compute an alignment of two sequences that corresponds to the least costly set of such mutations. Such an alignment may be thought of as minimizing the evolutionary distance or maximizing the similarity between the two sequences compared. In either case, the cost of this alignment is a measure of similarity; the algorithm guarantees it is optimal, based on the given scores. Because of their computational requirements, dynamic programming algorithms are impractical for searching large databases without the use of a supercomputer (Gotoh & Tagashira, 1986) or other special purpose hardware (Coulson *et al.*, 1987).

Rapid heuristic algorithms that attempt to approximate the above methods have been developed (Waterman, 1984), allowing large databases to be searched on commonly available computers. In many heuristic methods the measure of similarity is not explicitly defined as a minimal cost set of mutations, but instead is implicit in the algorithm itself. For example, the FASTP program (Lipman & Pearson, 1985; Pearson & Lipman, 1988) first finds locally similar regions between two sequences based on identities but not gaps, and then rescores these regions using a measure of similarity between residues, such as a PAM matrix (Dayhoff *et al.*, 1978) which allows conservative replacements as well as identities to increment the similarity score. Despite their rather indirect approximation of minimal evolution measures, heuristic tools such as FASTP have been quite popular and have identified many distant but biologically significant relationships.

0022-2836/90/190403-08 $03.00/0

In this paper we describe a new method, BLAST† (Basic Local Alignment Search Tool), which employs a measure based on well-defined mutation scores. It directly approximates the results that would be obtained by a dynamic programming algorithm for optimizing this measure. The method will detect weak but biologically significant sequence similarities, and is more than an order of magnitude faster than existing heuristic algorithms.

2. Methods

(a) *The maximal segment pair measure*

Sequence similarity measures generally can be classified as either global or local. Global similarity algorithms optimize the overall alignment of two sequences, which may include large stretches of low similarity (Needleman & Wunsch, 1970). Local similarity algorithms seek only relatively conserved subsequences, and a single comparison may yield several distinct subsequence alignments; unconserved regions do not contribute to the measure of similarity (Smith & Waterman, 1981; Goad & Kanehisa, 1982; Sellers, 1984). Local similarity measures are generally preferred for database searches, where cDNAs may be compared with partially sequenced genes, and where distantly related proteins may share only isolated regions of similarity, e.g. in the vicinity of an active site.

Many similarity measures, including the one we employ, begin with a matrix of similarity scores for all possible pairs of residues. Identities and conservative replacements have positive scores, while unlikely replacements have negative scores. For amino acid sequence comparisons we generally use the PAM-120 matrix (a variation of that of Dayhoff *et al.*, 1978), while for DNA sequence comparisons we score identities +5, and mismatches −4; other scores are of course possible. A sequence segment is a contiguous stretch of residues of any length, and the similarity score for two aligned segments of the same length is the sum of the similarity values for each pair of aligned residues.

Given these rules, we define a maximal segment pair (MSP) to be the highest scoring pair of identical length segments chosen from 2 sequences. The boundaries of an MSP are chosen to maximize its score, so an MSP may be of any length. The MSP score, which BLAST heuristically attempts to calculate, provides a measure of local similarity for any pair of sequences. A molecular biologist, however, may be interested in all conserved regions shared by 2 proteins, not only in their highest scoring pair. We therefore define a segment pair to be locally maximal if its score cannot be improved either by extending or by shortening both segments (Sellers, 1984). BLAST can seek all locally maximal segment pairs with scores above some cutoff.

Like many other similarity measures, the MSP score for 2 sequences may be computed in time proportional to the product of their lengths using a simple dynamic programming algorithm. An important advantage of the MSP measure is that recent mathematical results allow the statistical significance of MSP scores to be estimated under an appropriate random sequence model (Karlin & Altschul, 1990; Karlin *et al.*, 1990). Furthermore, for any particular scoring matrix (e.g. PAM-120) one can estimate the frequencies of paired residues in maximal segments. This tractability to mathematical analysis is a crucial feature of the BLAST algorithm.

(b) *Rapid approximation of MSP scores*

In searching a database of thousands of sequences, generally only a handful, if any, will be homologous to the query sequence. The scientist is therefore interested in identifying only those sequence entries with MSP scores over some cutoff score S. These sequences include those sharing highly significant similarity with the query as well as some sequences with borderline scores. This latter set of sequences may include high scoring random matches as well as sequences distantly related to the query. The biological significance of the high scoring sequences may be inferred almost solely on the basis of the similarity score, while the biological context of the borderline sequences may be helpful in distinguishing biologically interesting relationships.

Recent results (Karlin & Altschul, 1990; Karlin *et al.*, 1990) allow us to estimate the highest MSP score S at which chance similarities are likely to appear. To accelerate database searches, BLAST minimizes the time spent on sequence regions whose similarity with the query has little chance of exceeding this score. Let a word pair be a segment pair of fixed length w. The main strategy of BLAST is to seek only segment pairs that contain a word pair with a score of at least T. Scanning through a sequence, one can determine quickly whether it contains a word of length w that can pair with the query sequence to produce a word pair with a score greater than or equal to the threshold T. Any such hit is extended to determine if it is contained within a segment pair whose score is greater than or equal to S. The lower the threshold T, the greater the chance that a segment pair with a score of at least S will contain a word pair with a score of at least T. A small value for T, however, increases the number of hits and therefore the execution time of the algorithm. Random simulation permits us to select a threshold T that balances these considerations.

(c) *Implementation*

In our implementations of this approach, details of the 3 algorithmic steps (namely compiling a list of high-scoring words, scanning the database for hits, and extending hits) vary somewhat depending on whether the database contains proteins or DNA sequences. For proteins, the list consists of all words (w-mers) that score at least T when compared to some word in the query sequence. Thus, a query word may be represented by no words in the list (e.g. for common w-mers using PAM-120 scores) or by many. (One may, of course, insist that every w-mer in the query sequence be included in the word list, irrespective of whether pairing the word with itself yields a score of at least T.) For values of w and T that we have found most useful (see below), there are typically of the order of 50 words in the list for every residue in the query sequence, e.g. 12,500 words for a sequence of length 250. If a little care is taken in programming, the list of words can be generated in time essentially proportional to the length of the list.

The scanning phase raised a classic algorithmic problem, i.e. search a long sequence for all occurrences of certain short sequences. We investigated 2 approaches. Simplified, the first works as follows. Suppose that $w = 4$ and map each word to an integer between 1 and 20^4, so a

† Abbreviations used: BLAST, blast local alignment search tool; MSP, maximal segment pair; bp, base-pair(s).

word can be used as an index into an array of size $20^4 = 160{,}000$. Let the ith entry of such an array point to the list of all occurrences in the query sequence of the ith word. Thus, as we scan the database, each database word leads us immediately to the corresponding hits. Typically, only a few thousand of the 20^4 possible words will be in this table, and it is easy to modify the approach to use far fewer than 20^4 pointers.

The second approach we explored for the scanning phase was the use of a deterministic finite automaton or finite state machine (Mealy, 1955; Hopcroft & Ullman, 1979). An important feature of our construction was to signal acceptance on transitions (Mealy paradigm) as opposed to on states (Moore paradigm). In the automaton's construction, this saved a factor in space and time roughly proportional to the size of the underlying alphabet. This method yielded a program that ran faster and we prefer this approach for general use. With typical query lengths and parameter settings, this version of BLAST scans a protein database at approximately 500,000 residues/s.

Extending a hit to find a locally maximal segment pair containing that hit is straightforward. To economize time, we terminate the process of extending in one direction when we reach a segment pair whose score falls a certain distance below the best score found for shorter extensions. This introduces a further departure from the ideal of finding guaranteed MSPs, but the added inaccuracy is negligible, as can be demonstrated by both experiment and analysis (e.g. for protein comparisons the default distance is 20, and the probability of missing a higher scoring extension is about 0.001).

For DNA, we use a simpler word list, i.e. the list of all contiguous w-mers in the query sequence, often with $w = 12$. Thus, a query sequence of length n yields a list of $n-w+1$ words, and again there are commonly a few thousand words in the list. It is advantageous to compress the database by packing 4 nucleotides into a single byte, using an auxiliary table to delimit the boundaries between adjacent sequences. Assuming $w \geq 11$, each hit must contain an 8-mer hit that lies on a byte boundary. This observation allows us to scan the database byte-wise and thereby increase speed 4-fold. For each 8-mer hit, we check for an enclosing w-mer hit: if found, we extend as before. Running on a SUN4, with a query of typical length (e.g. several thousand bases), BLAST scans at approximately 2×10^6 bases/s. At facilities which run many such searches a day, loading the compressed database into memory once in a shared memory scheme affords a substantial saving in subsequent search times.

It should be noted that DNA sequences are highly nonrandom, with locally biased base composition (e.g. A+T-rich regions), and repeated sequence elements (e.g. *Alu* sequences) and this has important consequences for the design of a DNA database search tool. If a given query sequence has, for example, an A+T-rich subsequence, or a commonly occurring repetitive element, then a database search will produce a copious output of matches with little interest. We have designed a somewhat *ad hoc* but effective means of dealing with these 2 problems. The program that produces the compressed version of the DNA database tabulates the frequencies of all 8-tuples. Those occurring much more frequently than expected by chance (controllable by parameter) are stored and used to filter "uninformative" words from the query word list. Also, preceding full database searches, a search of a sublibrary of repetitive elements is performed, and the locations in the query of significant matches are stored. Words generated by these regions are removed from the query word list for the full search. Matches to the sublibrary, however, are reported in the final output. These 2 filters allow alignments to regions with biased composition, or to regions containing repetitive elements to be reported, as long as adjacent regions not containing such features share significant similarity to the query sequence.

The BLAST strategy admits numerous variations. We implemented a version of BLAST that uses dynamic programming to extend hits so as to allow gaps in the resulting alignments. Needless to say, this greatly slows the extension process. While the sensitivity of amino acid searches was improved in some cases, the selectivity was reduced as well. Given the trade-off of speed and selectivity for sensitivity, it is questionable whether the gap version of BLAST constitutes an improvement. We also implemented the alternative of making a table of all occurrences of the w-mers in the database, then scanning the query sequence and processing hits. The disk space requirements are considerable, approximately 2 computer words for every residue in the database. More damaging was that for query sequences of typical length, the need for random access into the database (as opposed to sequential access) made the approach slower, on the computer systems we used, than scanning the entire database.

3. Results

To evaluate the utility of our method, we describe theoretical results about the statistical significance of MSP scores, study the accuracy of the algorithm for random sequences at approximating MSP scores, compare the performance of the approximation to the full calculation on a set of related protein sequences and, finally, demonstrate its performance comparing long DNA sequences.

(a) *Performance of BLAST with random sequences*

Theoretical results on the distribution of MSP scores from the comparison of random sequences have recently become available (Karlin & Altschul, 1990; Karlin *et al.*, 1990). In brief, given a set of probabilities for the occurrence of individual residues, and a set of scores for aligning pairs of residues, the theory provides two parameters λ and K for evaluating the statistical significance of MSP scores. When two random sequences of lengths m and n are compared, the probability of finding a segment pair with a score greater than or equal to S is:

$$1 - e^{-y}, \tag{1}$$

where $y = Kmn\ e^{-\lambda S}$. More generally, the probability of finding c or more distinct segment pairs, all with a score of at least S, is given by the formula:

$$1 - e^{-y} \sum_{i=0}^{c-1} \frac{y^i}{i!}. \tag{2}$$

Using this formula, two sequences that share several distinct regions of similarity can sometimes be detected as significantly related, even when no segment pair is statistically significant in isolation.

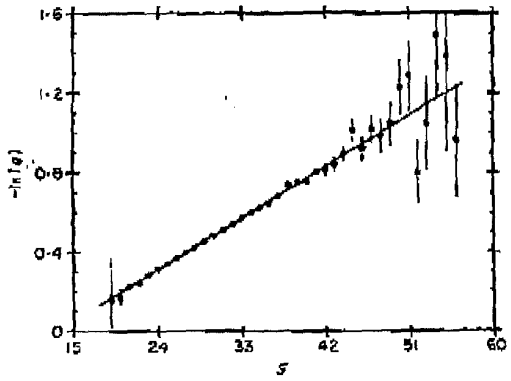


Figure 1. The probability q of BLAST missing a random maximal segment pair as a function of its score S.

While finding an MSP with a p-value of 0·001 may be surprising when two specific sequences are compared, searching a database of 10,000 sequences for similarity to a query sequence is likely to turn up ten such segment pairs simply by chance. Segment pair p-values must be discounted accordingly when the similar segments are discovered through blind database searches. Using formula (1), we can calculate the approximate score an MSP must have to be distinguishable from chance similarities found in a database.

We are interested in finding only segment pairs with a score above some cutoff S. The central idea of the BLAST algorithm is to confine attention to segment pairs that contain a word pair of length w with a score of at least T. It is therefore of interest to know what proportion of segment pairs with a given score contain such a word pair. This question makes sense only in the context of some distribution of high-scoring segment pairs. For MSPs arising from the comparison of random sequences, Dembo & Karlin (1991) provide such a limiting distribution. Theory does not yet exist to calculate the probability q that such a segment pair will fail to contain a word pair with a score of at least T. However, one argument suggests that q should depend exponentially upon the score of the MSP. Because the frequencies of paired letters in MSPs approaches a limiting distribution (Karlin & Altschul, 1990), the expected length of an MSP grows linearly with its score. Therefore, the longer an MSP, the more independent chances it effectively has for containing a word with a score of at least T, implying that q should decrease exponentially with increasing MSP score S.

To test this idea, we generated one million pairs of "random protein sequences" (using typical amino acid frequencies) of length 250, and found the MSP for each using PAM-120 scores. In Figure 1, we plot the logarithm of the fraction q of MSPs with score S that do not contain a word pair of length four with score at least 18. Since the values shown are subject to statistical variation, error bars represent one standard deviation. A regression line is plotted, allowing for heteroscedasticity (differing degrees of accuracy of the q-values). The correlation coefficient for $-\ln(q)$ and S is 0·999, suggesting that for practical purposes our model of the exponential dependence of q upon S is valid.

We repeated this analysis for a variety of word lengths and associated values of T. Table 1 shows the regression parameters a and b found for each instance: the correlation coefficient was always greater than 0·995. Table 1 also shows the implied percentage $q = e^{-(aS+b)}$ of MSPs with various scores that would be missed by the BLAST algorithm. These numbers are of course properly applicable only to chance MSPs. However, using a log-odds score matrix such as the PAM-120 that is based upon empirical studies of homologous proteins, high-scoring chance MSPs should resemble MSPs that reflect true homology (Karlin & Altschul, 1990). Therefore, Table 1 should provide a rough guide to the performance of BLAST on homologous as well as chance MSPs.

Based on the results of Karlin *et al.* (1990), Table 1 also shows the expected number of MSPs found when searching a random database of 16,000 length 250 protein sequences with a length 250 query. (These numbers were chosen to approximate the current size of the PIR database and the length of an average protein.) As seen from Table 1, only MSPs with a score over 55 are likely to be distinguishable from chance similarities. With $w = 4$ and $T = 17$, BLAST should miss only about a fifth of the MSPs with this score, and only about a tenth of MSPs with a score near 70. We will consider below the algorithm's performance on real data.

(b) *The choice of word length and threshold parameters*

On what basis do we choose the particular setting of the parameters w and T for executing BLAST on real data? We begin by considering the word length w.

The time required to execute BLAST is the sum of the times required (1) to compile a list of words that can score at least T when compared with words from the query; (2) to scan the database for hits (i.e. matches to words on this list); and (3) to extend all hits to seek segment pairs with scores exceeding the cutoff. The time for the last of these tasks is proportional to the number of hits, which clearly depends on the parameters w and T. Given a random protein model and a set of substitution scores, it is simple to calculate the probability that two random words of length w will have a score of at least T, i.e. the probability of a hit arising from an arbitrary pair of words in the query and the database. Using the random model and scores of the previous section, we have calculated these probabilities for a variety of parameter choices and recorded them in Table 1. For a given level of sensitivity (chance of missing an MSP), one can ask what choice of w minimizes the

Table 1

The probability of a hit at various settings of the parameters w and T, and the proportion of random MSPs missed by BLAST

			Linear regression $-\ln(q) \approx aS+b$		Implied % of MSPs missed by BLAST when S equals						
w	T	Probability of a hit $\times 10^5$	a	b	45	50	55	60	65	70	75
3	11	253	0·1238	−1·005	1	1	0	0	0	0	0
	12	147	0·0975	−0·746	4	3	2	1	1	0	0
	13	89	0·0625	−0·579	11	8	6	4	3	2	2
	14	48	0·0403	−0·461	20	16	12	10	8	6	5
	15	26	0·0328	−0·353	33	28	23	20	17	14	12
	16	14	0·0232	−0·263	46	41	36	32	28	26	23
	17	7	0·0158	−0·191	59	55	51	47	43	40	37
	18	4	0·0109	−0·137	70	67	63	60	57	54	51
4	13	127	0·1192	−1·278	2	1	1	0	0	0	0
	14	78	0·0904	−1·012	5	3	2	1	1	0	0
	15	47	0·0686	−0·802	10	7	5	4	3	2	1
	16	28	0·0519	−0·634	18	14	11	8	6	5	4
	17	16	0·0390	−0·498	28	23	19	16	13	11	9
	18	9	0·0290	−0·387	40	35	30	26	22	19	17
	19	5	0·0215	−0·298	51	46	41	37	33	30	27
	20	3	0·0159	−0·234	62	57	53	49	45	41	38
5	15	64	0·1157	−1·525	3	2	1	1	0	0	0
	16	40	0·0882	−1·207	6	4	3	2	1	1	0
	17	25	0·0670	−0·939	12	9	6	4	3	2	2
	18	15	0·0529	−0·754	20	15	12	9	7	5	4
	19	9	0·0413	−0·608	29	23	19	15	13	10	8
	20	5	0·0327	−0·506	38	32	28	23	20	17	14
	21	3	0·0257	−0·420	48	42	37	32	28	25	22
	22	2	0·0200	−0·343	57	52	47	42	38	35	31
Expected no. of random MSPs with score at least S:					50	9	2	0·3	0·05	0·01	0·002

chance of a hit. Examining Table 1, it is apparent that the parameter pairs ($w = 3$, $T = 14$), ($w = 4$, $T = 16$) and ($w = 5$, $T = 18$) all have approximately equivalent sensitivity over the relevant range of cutoff scores. The probability of a hit yielded by these parameter pairs is seen to decrease for increasing w; the same also holds for different levels of sensitivity. This makes intuitive sense, for the longer the word pair examined the more information gained about potential MSPs. Maintaining a given level of sensitivity, we can therefore decrease the time spent on step (3), above, by increasing the parameter w. However, there are complementary problems created by large w. For proteins there are 20^w possible words of length w, and for a given level of sensitivity the number of words generated by a query grows exponentially with w. (For example, using the 3 parameter pairs above, a 30 residue sequence was found to generate word lists of size 296, 3561 and 40,939 respectively.) This increases the time spent on step (1), and the amount of memory required. In practice, we have found that for protein searches the best compromise between these considerations is with a word size of four; this is the parameter setting we use in all analyses that follow.

Although reducing the threshold T improves the approximation of MSP scores by BLAST, it also increases execution time because there will be more words generated by the query sequence and therefore more hits. What value of T provides a reasonable compromise between the considerations of sensitivity and time? To provide numerical data, we compared a random 250 residue sequence against the entire PIR database (Release 23·0, 14,372 entries and 3,977,903 residues) with T ranging from 20 to 13. In Figure 2 we plot the execution time (user time on a SUN4-280) *versus* the number of

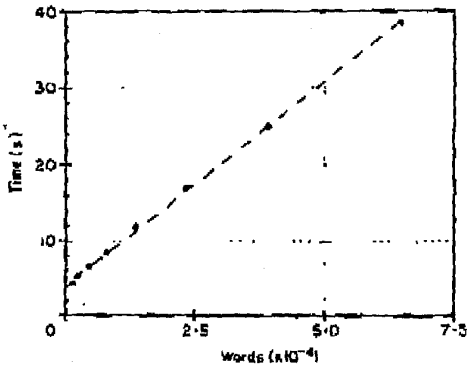


Figure 2. The central processing unit time required to execute BLAST on the PIR protein database (Release 23·0) as a function of the size of the word list generated. Points correspond to values of the threshold parameter T ranging from 13 to 20. Greater values of T imply fewer words in the list.

Table 2

The central processing unit time required to execute BLAST as a function of the approximate probability q of missing an MSP with score S

q (%)	CPU time (s)			
2	30	25	17	12
5	25	17	12	8
10	17	12	9	7
20	12	9	7	5
S:	44	55	70	90
p-value	10^{-1}	10^{-2}	10^{-3}	10^{-5}

Times are for searching the PIR database (Release 23·0) with a random query sequence of length 250 using a SUN4-280. CPU, central processing unit.

words generated for each value of T. Although there is a linear relationship between the number of words generated and execution time, the number of words generated increases exponentially with decreasing T over this range (as seen by the spacing of x values). This plot and a simple analysis reveal that the expected-time computational complexity of BLAST is approximately $aW+bN+cNW/20^w$, where W is the number of words generated, N is the number of residues in the database and a, b and c are constants. The W term accounts for compiling the word list, the N term covers the database scan, and the NW term is for extending the hits. Although the number of words generated, W, increases exponentially with decreasing T, it increases only linearly with the length of the query, so that doubling the query length doubles the number of words. We have found in practice that $T=17$ is a good choice for the threshold because, as discussed below, lowering the parameter further provides little improvement in the detection of actual homologies.

BLAST's direct tradeoff between accuracy and speed is best illustrated by Table 2. Given a specific probability q of missing a chance MSP with score S, one can calculate what threshold parameter T is required, and therefore the approximate execution time. Combining the data of Table 1 and Figure 2, Table 2 shows the central processing unit times required (for various values of q and S) to search the current PIR database with a random query sequence of length 250. To have about a 10% chance of missing an MSP with the statistically significant score of 70 requires about nine seconds of central processing unit time. To reduce the chance of missing such an MSP to 2% involves lowering T, thereby doubling the execution time. Table 2 illustrates, furthermore, that the higher scoring (and more statistically significant) an MSP, the less time is required to find it with a given degree of certainty.

(c) *Performance of BLAST with homologous sequences*

To study the performance of BLAST on real data, we compared a variety of proteins with other members of their respective superfamilies (Dayhoff, 1978), computing the true MSP scores as well as the BLAST approximation with word length four and various settings of the parameter T. Only with superfamilies containing many distantly related proteins could we obtain results usefully comparable with the random model of the previous section. Searching the globins with woolly monkey myoglobin (PIR code MYMQW), we found 178 sequences containing MSPs with scores between 50 and 80. Using word length four and T parameter 17, the random model suggests BLAST should miss about 24 of these MSPs; in fact, it misses 48. This poorer than expected performance is due to the uniform pattern of conservation in the globins, resulting in a relatively small number of high-scoring words between distantly related proteins. A contrary example was provided by comparing the mouse immunoglobulin κ chain precursor V region (PIR code KVMSTI) with immunoglobulin sequences, using the same parameters as previously. Of the 33 MSPs with scores between 45 and 65, BLAST missed only two; the random model suggests it should have missed eight. In general, the distribution of mutations along sequences has been shown to be more clustered than predicted by a Poisson process (Uzzell & Corbin, 1971), and thus the BLAST approximation should, on average, perform better on real sequences than predicted by the random model.

BLAST's great utility is for finding high-scoring MSPs quickly. In the examples above, the algorithm found all but one of the 89 globin MSPs with a score over 80, and all of the 125 immunoglobulin MSPs with a score over 50. The overall performance of BLAST depends upon the distribution of MSP scores for those sequences related to the query. In many instances, the bulk of the MSPs that are distinguishable from chance have a high enough score to be found readily by BLAST, even using relatively high values of the T parameter. Table 3 shows the number of MSPs with a score above a given threshold found by BLAST when searching a variety of superfamilies using a variety of T parameters. In each instance, the threshold S is chosen to include scores in the borderline region, which in a full database search would include chance similarities as well as biologically significant relationships. Even with T equal to 18, virtually all the statistically significant MSPs are found in most instances.

Comparing BLAST (with parameters $w=4$, $T=17$) to the widely used FASTP program (Lipman & Pearson 1985; Pearson & Lipman, 1988) in its most sensitive mode (*ktup* = 1), we have found that BLAST is of comparable sensitivity, generally yields fewer false positives (high-scoring but unrelated matches to the query), and is over an order of magnitude faster.

(d) *Comparison of two long DNA sequences*

Sequence data exist for a 73,360 bp section of the human genome containing the β-like globin gene

Table 3
The number of MSPs found by BLAST when searching various protein superfamilies in the PIR database (Release 22.0)

PIR code of query sequence	Superfamily searched	Cutoff score S	Number of MSPs with score at least S found by BLAST with T parameter set to							Number of MSPs in superfamily with score at least S
			22	20	19	18	17	16	15	
MYMQW	Globin	47	115	169	178	222	236	255	261	285
KVMST1	Immunoglobulin	47	153	155	155	156	156	157	158	158
OKBOG	Protein kinase	52	9	42	47	58	60	60	60	60
ITHU	Serpin	60	12	12	12	12	12	12	12	12
KYBOA	Serine protease	49	59	59	59	59	59	59	59	59
CCHU	Cytochrome c	46	81	91	91	96	98	98	98	96
FECF	Ferredoxin	44	22	23	23	24	24	24	24	24

MYMQW, woolly monkey myoglobin; KVMST1, mouse Ig κ chain precursor V region; OKBOG, bovine cGMP-dependent protein kinase; ITHU, human α-1-antitrypsin precursor; KYBOA, bovine chymotrypsinogen A; CCHU, human cytochrome c; FECF, *Chlorobium* sp. ferredoxin.

cluster and for a corresponding 44,595 bp section of the rabbit genome (Margot *et al.*, 1989). The pair exhibits three main classes of locally similar regions, namely genes, long interspersed repeats and certain anticipated weaker similarities, as described below. We used the BLAST algorithm to locate locally similar regions that can be aligned without introduction of gaps.

The human gene cluster contains six globin genes, denoted ε, $^{G}\gamma$, $^{A}\gamma$, η, δ and β, while the rabbit cluster has only four, namely ε, γ, δ and β. (Actually, rabbit δ is a pseudogene.) Each of the 24 gene pairs, one human gene and one rabbit gene, constitutes a similar pair. An alignment of such a pair requires insertion and deletions, since the three exons of one gene generally differ somewhat in their lengths from the corresponding exons of the paired gene, and there are even more extensive variations among the introns. Thus, a collection of the highest scoring alignments between similar regions can be expected to have at least 24 alignments between gene pairs.

Mammalian genomes contain large numbers of long interspersed repeat sequences, abbreviated *LINES*. In particular, the human β-like globin cluster contains two overlapped L1 sequences (a type of *LINE*) and the rabbit cluster has two tandem L1 sequences in the same orientation, both around 6000 bp in length. These human and rabbit L1 sequences are quite similar and their lengths make them highly visible in similarity computations. In all, eight L1 sequences have been cited in the human cluster and five in the rabbit cluster, but because of their reduced length and/or reversed orientation, the other published L1 sequences do not affect the results discussed below. Very recently, another piece of an L1 sequence has been discovered in the rabbit cluster (Huang *et al.*, 1990).

Evolution theory suggests that an ancestral gene cluster arranged as 5′-ε-γ-η-δ-β-3′ may have existed before the mammalian radiation. Consistent with this hypothesis, there are inter-gene similarities within the β clust rs. For example, there is a region between human ε and $^{G}\gamma$ that is similar to a region between rabbit ε and γ.

We applied a variant of the BLAST program to these two sequences, with match score 5, mismatch score −4 and, initially, $w = 12$. The program found 98 alignments scoring over 200, with 1301 being the highest score. Of the 57 alignments scoring over 350, 45 paired genes (with each of the 24 possible gene pairs represented) and the remaining 12 involved L1 sequences. Below 350, inter-gene similarities (as described above) appear, along with additional alignments of genes and of L1 sequences. Two alignments with scores between 200 and 350 do not fit the anticipated pattern. One reveals the newly discovered section of L1 sequence. The other aligns a region immediately 5′ from the human β gene with a region just 5′ from rabbit δ. This last alignment may be the result of an intrachromosomal gene conversion between δ and β in the rabbit genome (Hardison & Margot, 1984).

With smaller values of w, more alignments are found. In particular, with $w = 8$, an additional 32 alignments are found with a score above 200. All of these fall in one of the three classes discussed above. Thus, use of a smaller w provides no essentially new information. The dependence of various values on w is given in Table 4. Time is measured in seconds on a SUN4 for a simple variant of BLAST that works with uncompressed DNA sequences.

Table 4
The time and sensitivity of BLAST on DNA sequences as a function of w

w	Time	Words	Hits	Matches
8	15·9	44,587	118,941	130
9	6·8	44,586	39,218	123
10	4·3	44,585	15,321	114
11	3·6	44,584	7245	106
12	3·2	44,583	4197	98

4. Conclusion

The concept underlying BLAST is simple and robust and therefore can be implemented in a number of ways and utilized in a variety of contexts. As mentioned above, one variation is to allow for gaps in the extension step. For the applications we have had in mind, the tradeoff in speed proved unacceptable, but this may not be true for other applications. We have implemented a shared memory version of BLAST that loads the compressed DNA file into memory once, allowing subsequent searches to skip this step. We are implementing a similar algorithm for comparing a DNA sequence to the protein database, allowing translation in all six reading frames. This permits the detection of distant protein homologies even in the face of common DNA sequencing errors (replacements and frame shifts). C. B. Lawrence (personal communication) has fashioned score matrices derived from consensus pattern matching methods (Smith & Smith, 1990), and different from the PAM-120 matrix used here, which can greatly decrease the time of database searches for sequence motifs.

The BLAST approach permits the construction of extremely fast programs for database searching that have the further advantage of amenability to mathematical analysis. Variations of the basic idea as well as alternative implementations, such as those described above, can adapt the method for different contexts. Given the increasing size of sequence databases, BLAST can be a valuable tool for the molecular biologist. A version of BLAST in the C programming language is available from the authors upon request (write to W. Gish); it runs under both 4.2 BSD and the AT&T System V UNIX operating systems.

W.M. is supported in part by NIH grant LM05110, and E.W.M. is supported in part by NIH grant LM04960.

References

Coulson, A. F. W., Collins, J. F. & Lyall, A. (1987). *Comput. J.* 30, 420–424.

Dayhoff, M. O. (1978). Editor of *Atlas of Protein Sequence and Structure*, vol. 5, suppl. 3, Nat. Biomed. Res. Found., Washington, DC.

Dayhoff, M. O., Schwartz, R. M. & Orcutt, B. C. (1978). In *Atlas of Protein Sequence and Structure* (Dayhoff, M. O., ed.), vol. 5, suppl. 3, pp. 345–352, Nat. Biomed. Res. Found., Washington, DC.

Dembo, A. & Karlin, S. (1991). *Ann. Prob.* in the press.

Gaud, W. B. & Kanehisa, M. I. (1982). *Nucl. Acids Res.* 10, 247–263.

Gotoh, O. & Tagashira, Y. (1986). *Nucl. Acids Res.* 14, 57–64.

Hardison, R. C. & Margot, J. B. (1984). *Mol. Biol. Evol.* 1, 302–316.

Hopcroft, J. E. & Ullman, J. D. (1979). In *Introduction to Automata Theory, Languages, and Computation*, pp. 42–45, Addison-Wesley, Reading, MA.

Huang, X., Hardison, R. C. & Miller, W. (1990). *Comput. Appl. Biosci.* In the press.

Karlin, S. & Altschul, S. F. (1990). *Proc. Nat. Acad. Sci., U.S.A.* 87, 2264–2268.

Karlin, S., Dembo, A. & Kawabata, T. (1990). *Ann. Stat.* 18, 571–581.

Lipman, D. J. & Pearson, W. R. (1985). *Science*, 227, 1435–1441.

Margot, J. B., Demers, G. W. & Hardison, R. C. (1988). *J. Mol. Biol.* 203, 15–40.

Mealy, G. H. (1955). *Bell System Tech. J.* 34, 1045–1079.

Needleman, S. B. & Wunsch, C. D. (1970). *J. Mol. Biol.* 48, 443–453.

Pearson, W. R. & Lipman, D. J. (1988). *Proc. Nat. Acad. Sci., U.S.A.* 85, 2444–2448.

Sankoff, D. & Kruskal, J. B. (1983) *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley, Reading, MA.

Sellers, P. H. (1974). *SIAM J. Appl. Math.* 26, 787–793.

Sellers, P. H. (1984). *Bull. Math. Biol.* 46, 501–514.

Smith, R. F. & Smith, T. F. (1990). *Proc. Nat. Acad. Sci., U.S.A.* 87, 118–122.

Smith, T. F. & Waterman, M. S. (1981). *Advan. Appl. Math.* 2, 482–489.

Uzzell, T. & Corbin, K. W. (1971). *Science*, 172, 1089–1096.

Waterman, M. S. (1984). *Bull. Math. Biol.* 46, 473–500.

Edited by S. Brenner

J. Mol. Biol. (1991) 219, 555–565

Amino Acid Substitution Matrices from an Information Theoretic Perspective

Stephen F. Altschul

National Center for Biotechnology Information
National Library of Medicine
National Institutes of Health
Bethesda, MD 20894, U.S.A.

(Received 1 October 1990; accepted 12 February 1991)

Protein sequence alignments have become an important tool for molecular biologists. Local alignments are frequently constructed with the aid of a "substitution score matrix" that specifies a score for aligning each pair of amino acid residues. Over the years, many different substitution matrices have been proposed, based on a wide variety of rationales. Statistical results, however, demonstrate that any such matrix is implicitly a "log-odds" matrix, with a specific target distribution for aligned pairs of amino acid residues. In the light of information theory, it is possible to express the scores of a substitution matrix in bits and to see that different matrices are better adapted to different purposes. The most widely used matrix for protein sequence comparison has been the PAM-250 matrix. It is argued that for database searches the PAM-120 matrix generally is more appropriate, while for comparing two specific proteins with suspected homology the PAM-200 matrix is indicated. Examples discussed include the lipocalins, human α_1B-glycoprotein, the cystic fibrosis transmembrane conductance regulator and the globins.

Keywords: homology; sequence comparison; statistical significance; alignment algorithms; pattern recognition

1. Introduction

General methods for protein sequence comparison were introduced to molecular biology 20 years ago and have since gained widespread use. Most early attempts to measure protein sequence similarity focused on global sequence alignments, in which every residue of the two sequences compared had to participate (Needleman & Wunsch, 1970; Sellers, 1974; Sankoff & Kruskal, 1983). However, because distantly related proteins may share only isolated regions of similarity, e.g. in the vicinity of an active site, attention has shifted to local as opposed to global sequence similarity measures. The basic idea is to consider only relatively conserved subsequences; dissimilar regions do not contribute to or subtract from the measure of similarity. Local similarity may be studied in a variety of ways. These include measures based on the longest matching segments of two sequences with a specified number or proportion of mismatches (Arratia *et al.*, 1986; Arratia & Waterman, 1989), as well as methods that compare all segments of a fixed, predefined "window" length (McLachlan, 1971). The most common practice, however, is to consider segments of all lengths, and choose those that optimize a similarity measure (Smith & Waterman, 1981; Goad & Kanehisa, 1982; Sellers, 1984). This has the advantage of placing no *a priori* restrictions on the length of the local alignments sought. Most database search methods have been based on such local alignments (Lipman & Pearson, 1985; Pearson & Lipman, 1988; Altschul *et al.*, 1990).

To evaluate local alignments, scores generally are assigned to each aligned pair of residues (the set of such scores is called a substitution matrix), as well as to residues aligned with nulls; the score of the overall alignment is then taken to be the sum of these scores. Specifying an appropriate amino acid substitution matrix is central to protein comparison methods and much effort has been devoted to defining, analyzing and refining such matrices (McLachlan, 1971; Dayhoff *et al.*, 1978; Schwartz & Dayhoff, 1978; Feng *et al.*, 1985; Rao, 1987; Risler *et al.*, 1988). One hope has been to find a matrix best adapted to distinguishing distant evolutionary relationships from chance similarities. Recent mathematical results (Karlin & Altschul, 1990; Karlin *et al.*, 1990) allow all substitution matrices to be viewed in a common light, and provide a rationale for selecting particular sets of "optimal" scores for local protein sequence comparison.

0022-2836/91/110555-11 $03.00/0

2. The Statistical Significance of Local Sequence Alignments

Global alignments are of essentially no use unless they can allow gaps, but this is not true for local alignments. The ability to choose segments with arbitrary starting positions in each sequence means that biologically significant regions frequently may be aligned without the need to introduce gaps. While, in general, it is desirable to allow gaps in local alignments, doing so greatly decreases their mathematical tractability. The results described here apply rigorously only to local alignments that lack gaps, i.e. to segments of equal length from each of the two sequences compared. Some recent database search tools have focused on finding such alignments (Altschul & Lipman, 1990; Altschul *et al.*, 1990). However, the statistics of optimal scores for local alignments that include gaps (Smith *et al.*, 1985; Waterman *et al.*, 1987) are broadly analogous to those for the no-gap case (Karlin & Altschul, 1990; Karlin *et al.*, 1990), where more precise results are available. Therefore, one may hope that many of the basic ideas presented below will generalize to local alignments that include gaps.

Formally, we assume that the aligned amino acids a_i and a_j are assigned the substitution score s_{ij}. Given two protein sequences, the pair of equal length segments that, when aligned, have the greatest aggregate score we call the Maximal Segment Pair (MSP†). An MSP may be of any length; its score is the MSP score.

Since any two protein sequences, related or unrelated, will have some MSP score, it is important to know how great a score one can expect to find simply by chance. To address this question one needs some model of chance. The simplest is to assume that in the two proteins compared, the amino acid a_i appears randomly with the probability p_i. These probabilities are chosen to reflect the observed frequencies of the amino acids in actual proteins. For simplicity of discussion we will assume both proteins share the same amino acid probability distribution; more generally, one can allow them to have different distributions. A random protein sequence is simply one constructed according to this model.

For the sake of the statistical theory, we need to make two crucial but reasonable assumptions about the substitution scores. The first is that there be at least one positive score and the second is that the expected score $\sum_{i,j} p_i p_j s_{ij}$ be negative. Because we permit the length of a segment pair to be adjusted to optimize its score, both these assumptions are necessary also from a practical perspective. If there were no positive scores, the MSP would always consist of a single pair of residues (or none at all, if this were permitted), and such an alignment is not of interest. If the expected score for two random residues were positive, extending a segment pair as far as possible would always tend to increase its score; this violates the idea of seeking local alignments. Substitution matrices used in other contexts, such as global alignments (Needleman & Wunsch, 1970) or local alignments using windows (McLachlan, 1971), need not satisfy these constraints. However, unless otherwise stated, it will be assumed below that any substitution matrix satisfies the two conditions described.

The statistical theory of MSP scores (Karlin & Altschul, 1990; Karlin *et al.*, 1990) involves a key parameter λ, which is the unique positive solution to the equation:

$$\sum_{i,j} p_i p_j e^{\lambda s_{ij}} = 1. \qquad (1)$$

Notice that multiplying all the scores of a substitution matrix by some positive constant does not effect the relative scores of any subalignments. Two matrices related by such a factor can, therefore, be considered essentially equivalent. Inspection of equation (1) reveals that multiplying all scores by α also has the effect of dividing λ by α. The parameter λ may, therefore, be viewed as a natural scale for any scoring system: its deeper meaning will be discussed below.

Given two random protein sequences as described above, how many distinct, or "locally optimal" (Sellers, 1984) MSPs with score at least S are expected to occur simply by chance? This number is well approximated by the formula:

$$KN e^{-\lambda S} \qquad (2)$$

where N is the product of the sequences' lengths, and K is an explicitly calculable parameter (Karlin & Altschul, 1990; Karlin *et al.*, 1990). When comparing a single random sequence with all the sequences in a database, setting N to the product of the query sequence length and the database length (in residues) yields an upper bound on the number of distinct MSPs with score at least S that the search is expected to yield.

3. Optimal Substitution Matrices for Local Sequence Alignment

Formula (2) allows us to tell when a segment pair has a significantly high score. However, it does not assist in choosing an appropriate substitution matrix in the first place. A second class of results, however, has direct bearing on this question. These state that among MSPs from the comparison of random sequences, the amino acids a_i and a_j are aligned with frequency approaching $q_{ij} = p_i p_j e^{\lambda s_{ij}}$ (Arratia *et al.*, 1988; Karlin & Altschul, 1990; Karlin *et al.*, 1990; Dembo & Karlin, 1991).

Given any substitution matrix and random protein model, one may easily calculate the set of target frequencies, q_{ij}, just described. Notice that by the definition of λ in equation (1), these target frequencies sum to 1. Now among alignments representing distant homologies, the amino acids are

† Abbreviations used: MSP, Maximal Segment Pair; Ig, immunoglobulin.

paired with certain characteristic frequencies. Only if these correspond to a matrix's target frequencies, it has been argued, can the matrix be optimal for distinguishing distant local homologies from similarities due to chance (Karlin & Altschul, 1990).

Any substitution matrix has an implicit set of target frequencies for aligned amino acids. Writing the scores of the matrix in terms of its target frequencies, one has:

$$s_{ij} = \left(\ln \frac{q_{ij}}{p_i p_j}\right) \Big/ \lambda. \qquad (3)$$

In other words, the score for an amino acid pair can be written as the logarithm to some base of that pair's target frequency divided by the background frequency with which the pair occurs. Such a ratio compares the probability of an event occurring under two alternative hypotheses and is called a likelihood or odds ratio. Scores that are the logarithm of odds ratios are called log-odds scores. Adding such scores can be thought of as multiplying the corresponding probabilities, which is appropriate for independent events, so that the total score remains a log-odds score.

Log-odds matrices have been advocated in a number of contexts, (Dayhoff *et al.*, 1978; Gribskov *et al.*, 1987; Stormo & Hartzell, 1989). The widely used PAM matrices (Dayhoff *et al.*, 1978), for instance, are explicitly of this form. Other substitution matrices, though based on a wide variety of rationales, are all log-odds matrices, but with implicit rather than explicit target frequencies. Therefore, while one may criticize the method described by Dayhoff *et al.* for estimating appropriate target frequencies (Wilbur, 1985), the most direct way to derive superior matrices appears to be through the refined estimation of amino acid pair target and background frequencies rather than through any fundamentally different approach.

4. Substitution Matrices for Global Alignments

While we have been considering substitution matrices in the context of local sequence comparison, they may be employed for global alignment as well (Needleman & Wunsch, 1970; Sellers, 1974; Schwartz & Dayhoff, 1978). There is a fundamental difference, however, between the use of such matrices in these two contexts. For global alignments, as previously, multiplying all scores by a fixed positive number has no effect on the relative scores of different alignments. But adding a fixed quantity a to the score for aligning any pair of residues (and $a/2$ to the score for aligning a residue with a null) likewise has no effect. Scoring systems that may be transformed into one another by means of these two rules are, for all practical purposes, equivalent. Unfortunately, the new transformation means that no unique log-odds interpretation of global substitution matrices is possible, and it is doubtful that any "target distribution" theorem can be proved. It may be possible to make a convincing case for a particular substitution matrix in the global alignment context, but the argument will most likely have to be different from that for local alignments (Karlin & Altschul, 1990). The same applies to substitution matrices used with fixed-length windows for studying local similarities (McLachlan, 1971; Argos, 1987; Stormo & Hartzell, 1989): a fixed quantity can be added to all entries of such a matrix with no essential effect. It is notable that while the PAM matrices were developed originally for global sequence comparison (Dayhoff *et al.*, 1978), their statistical theory has blossomed in the local alignment context.

5. Local Alignment Scores as Measures of Information

Multiplying a substitution matrix by a constant changes λ but does not alter the matrix's implicit target frequencies. By appropriate scaling, one may therefore select the parameter λ at will. Writing the matrix in log-odds form, such scaling corresponds merely to using a different implicit base for the logarithm. One natural choice for λ is 1, so that all scores become natural logarithms. Perhaps more appealing is to choose $\lambda = \ln 2 \approx 0.693$, so that the base for the log-odds matrix becomes 2. This lends a particularly intuitive appeal to formula (2). Setting the expected number of MSPs with score at least S equal to p, and solving for S, one finds:

$$S = \log_2 \frac{K}{p} + \log_2 N. \qquad (4)$$

For typical substitution matrices, K is found to be near 0.1, and an alignment may be considered significant when p is 0.05. Therefore the right-hand side of equation (4) generally is dominated by the term $\log_2 N$. In other words, the score needed to distinguish an MSP from chance is approximately the number of bits needed to specify where the MSP starts in each of the two sequences being compared. (One bit can be thought of as the answer to a single yes-no question; it is the amount of information needed to distinguish between 2 possibilities. It becomes apparent that, in general, $\log_2 N$ bits of information are needed to distinguish among N possibilities.)

For comparing two proteins of length 250 amino acid residues, about 16 bits of information are required: for comparing one such protein to a sequence database containing 4,000,000 residues, about 30 bits are needed. When cast in this light, alignment scores are not arbitrary numbers. By appropriate scaling (multiplying by $\lambda/0.693$) they take on the units of bits, and rough significance calculations can be performed in one's head. Furthermore, when so normalized, different amino acid substitution matrices may be directly compared.

6. The Relative Entropy of a Substitution Matrix

The above review of previous results has provided us with the necessary tools for the analysis that follows. The ultimate goal is to decide which substitution matrices are the most appropriate for database searching and for detailed pairwise sequence comparison.

Given a random protein model and a substitution matrix, one may calculate the target frequencies q_{ij} characteristic of the alignments for which the matrix is optimized. A useful quantity to consider is the average score (information) per residue pair in these alignments. Assuming the substitution matrix is normalized as described above, this value is simply:

$$H = \sum_{i,j} q_{ij} s_{ij} = \sum_{i,j} q_{ij} \log_2 \frac{q_{ij}}{p_i p_j}. \quad (5)$$

Notice that H depends both on the substitution matrix and on the random protein model. In information theoretic terms, H is the relative entropy of the target and background distributions. The origin of the name need not be of concern. The important point is that, for an alignment characterized by the target frequencies q_{ij}, H measures the average information available per position to distinguish the alignment from chance. Intuitively, the higher the value of the relative entropy of target and background distributions, the more easily they are distinguished. For a high value of H, relatively short alignments with the target distribution can be distinguished from chance, while, if the value of H is lower, longer alignments are necessary.

It is interesting to examine the PAM model of molecular evolution (Dayhoff *et al.*, 1978) from this standpoint. From a study of mutations between a large number of closely related proteins, Dayhoff and co-workers proposed a stochastic model of protein evolution. The amount of evolutionary change that yields, on average, one substitution in 100 amino acid residues they called one PAM. Using their model, one may easily calculate the frequency with which any two amino acid residues are paired in an accurate alignment of two homologous proteins that have diverged by any given amount of evolutionary change. These target frequencies may then be used to construct log-odds matrices and, in particular, the widely used PAM-250 matrix. Dayhoff *et al.* (1978) originally proposed this matrix for the global alignment of two sequences suspected to be homologous, but it has since been used to search protein databases for local alignments to a query sequence (Lipman & Pearson, 1985; Pearson & Lipman, 1988). One may therefore inquire whether 250 PAMs yield reasonable target frequencies for database searches.

Assuming the model described by Dayhoff *et al.* (1978), Table 1 lists the relative entropy H implicit in a range of PAM matrices. As argued above, distinguishing an alignment from chance in a search of a typical current protein database using an average length protein requires about 30 bits of information. Accordingly, for an alignment of segments separated by a given PAM distance, one can calculate the minimum length necessary to rise above background noise; these lengths are recorded in Table 1. For instance, at a distance of 250 PAMs, on average only 0·36 bit of information is available per alignment position. To be statistically significant, such an alignment would need to have a length greater than about 83 residues. Many biologically interesting regions of protein similarity are much shorter than this, and accordingly need a stronger signal to be detected. A local alignment of length 20 residues will need about 1·5 bits per alignment position, while one of length 40 residues will need about 0·75 bit. Table 1 shows that such alignments will not be detectable if their constituent

Table 1

The relative entropy H of PAM matrices

PAM distance	H (bits)	Min. significant length (30 bits)	PAM distance	H (bits)	Min. significant length (30 bits)
0	4·17	8	180	0·60	51
10	3·43	9	190	0·55	55
20	2·85	11	200	0·51	59
30	2·57	12	210	0·48	63
40	2·26	14	220	0·45	68
50	2·00	15	230	0·42	73
60	1·79	17	240	0·39	78
70	1·60	19	250	0·36	83
80	1·44	21	260	0·34	89
90	1·30	24	270	0·32	94
100	1·18	26	280	0·30	100
110	1·08	28	290	0·28	107
120	0·98	31	300	0·27	113
130	0·90	34	310	0·25	120
140	0·82	37	320	0·24	127
150	0·76	40	330	0·22	134
160	0·70	43	340	0·21	141
170	0·65	47	350	0·20	149

Table 2

The average score (in bits) per alignment position when using given PAM matrices to compare segments in fact separated by a variety of PAM distances

PAM matrix *M* employed	Actual PAM distance *D* of segments							
	40	80	120	160	200	240	280	320
40	2·26	1·31	0·62	0·10	−0·30	−0·61	−0·85	−1·06
80	2·14	1·44	0·92	0·53	0·23	−0·02	−0·21	−0·37
120	1·93	1·39	0·98	0·67	0·42	0·22	0·06	−0·07
160	1·71	1·28	0·95	0·70	0·50	0·33	0·20	0·09
200	1·51	1·16	0·90	0·68	0·51	0·36	0·26	0·17
240	1·32	1·05	0·82	0·65	0·51	0·39	0·29	0·21
280	1·17	0·94	0·75	0·60	0·48	0·38	0·30	0·23
320	1·03	0·84	0·68	0·56	0·45	0·37	0·30	0·24

segments have diverged by more than about 75 and 150 PAMs, respectively.

7. PAM Matrices for Database Searching and Two-sequence Comparison

The relative entropy associated with a specific PAM distance indicates how much information per position is optimally available. For a given alignment, one can attain such a score only by using the appropriate PAM matrix, but, of course, before the alignment is found it will not be known which matrix that is. It has therefore been proposed that a variety of PAM matrices be used for database searches (Collins *et al.*, 1988). We seek here to analyze how many such matrices are necessary, and which should be used.

Suppose one uses a matrix optimized for PAM distance M to compare two homologous protein segments that are actually separated by PAM distance D. For a range of values of M and D, the average score achieved per alignment position is shown in Table 2. Notice that for any given matrix M, the smaller the actual distance D, the higher the score. On the other hand, for a specific distance D, the highest score corresponds to the matrix with PAM distance $M = D$; this score is just the relative entropy discussed above. Using a PAM matrix with M near D, however, can yield a near-optimal score.

Table 3

Ranges of local alignment lengths for which various PAM matrices are appropriate

PAM matrix	93% efficiency range for database searching (30 bits)	87% efficiency range for 2-sequence comparison (16 bits)
40	9 to 21	4 to 14
80	13 to 34	6 to 22
120	19 to 50	9 to 33
160	26 to 70	12 to 46
200	36 to 94	16 to 62
240	47 to 123	21 to 80
280	60 to 155	27 to 101
320	75 to 192	34 to 124
360	94 to 238	42 to 149

For example, the relative entropy for $D = 160$ is 0·70 bit, but any PAM matrix in the range 120 to 200 yields at least 0·67 bit per position. In practice, how near the optimal is it important to be?

As argued above, for a given PAM distance there is a critical length at which alignments are just distinguishable from chance in a typical current database search; these lengths are recorded in Table 1. For the sake of analysis, we will assume that it is worth performing an extra search (using a different PAM matrix) only if it is able to increase the score for such a critical alignment by about two bits, corresponding to a factor of 4 in significance. Since a critical alignment has about 30 bits of information, we will therefore be satisfied using a PAM matrix that yields a score greater than 93% of the optimal achievable. Using data such as those shown in Table 2, one can calculate for which PAM distances D (and thus for which critical lengths) a given matrix M is appropriate; the results are recorded in Table 3. Our experience has shown that perhaps the most typical lengths for distant local alignments are those for which the PAM-120 matrix gives near-optimal scores, i.e. lengths 19 to 50 residues. Therefore, if one wishes to use a single standard matrix for database searches, the PAM-120 matrix (Table 4) is a reasonable choice. This matrix may, however, miss short but strong or long but weak similarities that contain sufficient information to be found. Accordingly, Table 3 shows that to complement the PAM-120 matrix, the PAM-40 and PAM-240 (or traditional PAM-250) matrices can be used. Additional matrices should improve the detection of distant similarities only marginally (i.e. raise their scores by at most 2 bits).

If, rather than searching a database with a query sequence, one wishes to compare two specific sequences for which one already has evidence of relatedness, the background noise is greatly decreased. As discussed above, for two proteins of typical length, about 16 bits are needed to distinguish a local alignment from chance. Accordingly, applying the same criteria as before, a matrix should be considered adequate for those PAM distances at which it yields an average score within 87% of the optimal. In Table 3, we list the range of critical lengths over which various PAM

Table 4

The PAM-120 matrix with scores in half bits

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	3																			
R	−3	6																		
N	0	−1	4																	
D	0	−3	2	5																
C	−3	−4	−5	−7	9															
Q	−1	1	0	1	−7	6														
E	0	−3	1	3	−7	2	5													
G	1	−4	0	0	−5	−3	−1	5												
H	−3	1	2	0	−4	3	−1	−4	7											
I	−1	−2	−2	−3	−3	−3	−3	−4	−4	6										
L	−3	−4	−4	−5	−7	−2	−4	−5	−3	1	5									
K	−2	2	1	−1	−7	0	−1	−3	−2	−2	−4	5								
M	−2	−1	−3	−4	−6	−1	−4	−4	−4	1	3	0	8							
F	−4	−4	−4	−7	−6	−6	−6	−5	−2	0	0	−6	−1	8						
P	1	−1	−2	−2	−3	0	−1	−2	−1	−3	−3	−2	−3	−5	6					
S	1	−1	1	0	−1	−2	−1	1	−2	−2	−4	−1	−2	−3	1	3				
T	1	−2	0	−1	−3	−2	−2	−1	−3	0	−3	−1	−1	−4	−1	2	4			
W	−7	1	−5	−8	−8	−6	−8	−8	−5	−7	−5	−5	−7	−1	−7	−2	−6	12		
Y	−4	−6	−2	−5	−1	−5	−4	−6	−1	−2	−3	−6	−4	4	−6	−3	−3	−1	8	
V	0	−3	−3	−3	−2	−3	−3	−2	−3	3	1	−4	1	−3	−2	−2	0	−8	−3	5

matrices are appropriate for detailed pairwise sequence comparison. As a single matrix, the PAM-200 spans the most typical range of local alignment lengths, i.e. 16 to 62 residues. Alternatively, if two different matrices are to be used, the PAM-60 and PAM-250, which together span alignment lengths 6 to 85 residues, or the PAM-120 and PAM-320 matrices, which span lengths 9 to 124 residues, appear to be appropriate pairs.

Since it is convenient to express substitution matrices as integers, and since a probability factor of 2 between score levels is too rough, the units for the PAM-120 matrix shown in Table 4 are half bits. The scores in the original PAM-250 matrix (Dayhoff *et al.*, 1978) were scaled as $10 \times \log_{10}$. Because $10/(\ln 10) \approx 3/(\ln 2)$ to within 0·4%, a unit score in that matrix can be thought of as approximately one-third of a bit.

8. Biological Examples

As discussed, the particular PAM matrix that best distinguishes distant homologies from chance similarities found in a database search depends on the nature of the homologies present, and this cannot be known *a priori*. However, it is frequently the case that distantly related proteins will share isolated stretches of relatively conserved amino acid residues, corresponding to active sites or other important structural features. It has been observed that in general the mutations along genes coding for proteins are not Poisson-distributed (Uzzell & Corbin, 1971; Holmquist *et al.*, 1983), suggesting that short, conserved regions are to be expected. As shown in Table 3, this means that the widely used PAM-250 matrix generally will not be optimal for locating distant relationships.

In the examples below, we compare the PAM-250 and PAM-120 scores for MSPs representing distant relationships to four different query sequences. In all cases, we consider relationships near the limits of what can be distinguished from chance in a search of the PIR protein sequence database (Release 26·0; 7,348,950 residues). It will be noticed that the highest chance PAM-250 scores are consistently slightly smaller than the highest chance PAM-120 scores. This is primarily attributable to the fact that the parameter K discussed above is about half as large for the former scores as for the latter. Furthermore, since neither the PIR database nor a given query sequence ever precisely fits the random protein model described by Dayhoff *et al.* (1978), the parameter λ varies slightly from one comparison to another. Therefore, while we will treat the PAM-120 scores from Table 4 as half bits, and the PAM-250 scores of Dayhoff *et al.* (1978) as one-third bits, it should be noted that this is always a slight approximation.

(a) *Lipocalins*

We used the BLAST program (Altschul *et al.*, 1990) to search the PIR database with human apolipoprotein D precursor (PIR code LPHUD; Drayna *et al.*, 1987), using both the PAM-250 (Dayhoff *et al.*, 1978) and PAM-120 (Table 4) substitution matrices. Human apolipoprotein D precursor is a 189 residue glycoprotein that belongs to the lipocalin (α_2-microglobulin) superfamily, which contains proteins that exhibit a wide range of functions related to their ability to bind small hydrophobic ligands. The similarities among these proteins and their biological roles have been analyzed (Peitsch & Boguski, 1990), and crystal structures are available for several members of the superfamily (Cowan *et al.*, 1990). Three proteins in the superfamily are rat androgen-dependent epididymal protein (PIR code

Table 5

Three MSPs representing distant relationships, from searches of the PIR protein sequence database (release 26·0) with human apolipoprotein D precursor (PIR code LPHUD)

PIR code	Optimal PAM-250 alignment			Optimal PAM-250 score (bits)	Optimal PAM-120 score (bits)
LPHUD	25	LGKCPNPPVQENFDVNKYLGRWYEI	49		
SQRTAD	12	LAAGTEGAVVKDFDISKFLGFWYEI	36	27.0	33.5
A32202	27	HDTVQPNFQQDKFLGRWY	44	25.7	33.5
HCHU	28	NIQVQENFNISRIYGKWYNL	47	23.0	30.5
Highest chance alignment score:				27.0	29.0
PIR code of sequence involved:				S00758	S00758

LPHUD, human apolipoprotein D precursor; SQRTAD, rat androgen-dependent epididymal 18·5 K protein precursor; A32202, rat prostaglandin-D synthase; HCHU, human α_1-microglobulin/inter-α-trypsin inhibitor precursor; S00758, human surface glycoprotein CD16 precursor.

SQRTAD; Brooks *et al.*, 1986), rat prostaglandin-D synthase (PIR code A32202; Urade *et al.*, 1989) and human α_1-microglobulin (PIR code HCHU; Kaumeyer *et al.*, 1986). The second of these has only recently been recognized as a member of the superfamily (M. S. Boguski & M. C. Peitsch, personal communication); it is the first such member with known catalytic activity (Urade *et al.*, 1989).

Using PAM-250 scores, the maximal segment pair for each of these sequences when compared to LPHUD is shown in Table 5. These local similarities correspond to one of two motifs that are conserved throughout the superfamily (Boguski & States, 1990). The scores for the three alignments are 27·0, 25·7 and 23·0 bits, respectively. However, the highest score from a protein in the database unrelated to LPHUD is 27·0 bits, involving human surface glycoprotein CD16 precursor (PIR code S00758; Simmons & Seed, 1988). The PAM-250 matrix therefore fails to separate the homologous alignments shown from background noise. In contrast, using the PAM-120 matrix of Table 4, the scores for the three alignments jump to 33·5, 33·5 and 30·5 bits, respectively. (The 1st 7 alignment positions for LPHUD-SQRTAD shown in Table 5 are dropped in an optimal PAM-120 alignment, as are the 1st 3 positions for the LPHUD-A32202 alignment.) This raises their scores above that of the best chance PAM-120 alignment (29·0 bits), again involving human surface glycoprotein CD16 precursor. Notice that in both cases the estimate that about 30 bits are needed clearly to distinguish an MSP from chance is valid. For this query sequence, no relationship is found using the PAM-250 matrix that is missed by the PAM-120.

(b) *Human α_1B-glycoprotein*

We searched the PIR database with human α_1B-glycoprotein (PIR code OMHU1B; Ishioka *et al.*, 1986), a plasma glycoprotein of unknown function, and a member of the immunoglobulin superfamily. Using the PAM-250 matrix, the only protein in the database with an MSP that rises above background noise is pig Po2 F protein (PIR code PL0030; Van de Weghe *et al.*, 1988), which achieves a score of 32·3 bits. As shown in Table 6, the score for this known homology (Van de Weghe *et al.*, 1988) rises to 45·0 bits when the PAM-120 matrix is used instead. In addition, two proteins with immunoglobulin domains, kinase-related transforming protein precursor (PIR code S00474; Qiu *et al.*, 1988) and human Ig κ chain precursor V-III region (PIR code K3HUVH; Pech & Zachau, 1984), achieve scores of 29·0 and 28·5 bits, respectively. Table 6 illustrates that both these similarities are only just distinguishable from chance, and that using the PAM-250 matrix both similarities drop in score by at least four bits.

(c) *The cystic fibrosis transmembrane conductance regulator*

The cause of cystic fibrosis has been traced to mutations in a protein that bears striking similarity to many proteins involved in the transport of substances across the cell membrane (PIR code A30300; Riordan *et al.*, 1989). Characteristic features of the protein are two nucleotide (ATP)-binding folds (Higgins *et al.*, 1986). When the PIR database is searched with A30300, many related

Table 6

Three MSPs representing distant relationships, from searches of the PIR protein sequence database (release 26·0) with human α_1B-glycoprotein (PIR code OMHU1B)

PIR code	Optimal PAM-250 alignment			Optimal PAM-250 score (bits)	Optimal PAM-120 score (bits)
OMHU1B	1 AIFTETQPSLWAESESLLKPLANVTLTCQA 30				
PL0030	1 ALFLDPFPNLWAERQSLLEPWANVTLYSQS 30			32.3	45.0
OMHU1B	171 LSEPSATVTIEELAAPPFFVIMSHGESSQVLSFGNKVTLTCVAPLS 216				
S00474	18 LRGQTATSQPSASPGEPSPPSIHPAQSELIVEAGDTLSLTCIDP		61	25.0	29.0
K3HUVH	15 LPDTTREIVMTQSPPTLSLSPGERVTLSCRASQS		48	22.0	28.5
	Highest chance alignment score:			27.0	28.0
	PIR code of sequence involved:			JQ0102	WGSMHH

OMHU1B, human α_1B-glycoprotein; PL0030, pig Po2 F protein; S00474, kinase-related transforming protein (kit) precursor; K3HUVH, human Ig κ chain precursor V-III region (Vh); JQ0102, eggplant mosaic virus RNA replicase (Osorio-Keese *et al.*, 1989); WGSMHH, *Streptomyces hygromycin* B phosphotransferase (Zalacain *et al.*, 1986).

proteins may be identified easily using either the PAM-250 or the PAM-120 substitution matrix. However, several distant relationships present are harder to detect. In Table 7 are shown four optimal PAM-250 alignments, representing homologies to each of the two A30300 nucleotide-binding folds. None of these alignments has a PAM-250 score as great as the highest chance score of 31·3 bits. In contrast, when the PAM-120 matrix is used, the alignments jump in score by 4 to almost 12 bits, giving all but one a score greater than the highest chance PAM-120 score of 33·0 bits. (The boundaries of the optimal alignments change slightly under the alternate scoring scheme.) No biologically significant similarity is distinguished by the PAM-250 matrix that is not found using the PAM-120. The relatively high chance scores found in this example are partly attributable to the length of the query

Table 7

Four MSPs representing distant relationships, from searches of the PIR protein sequence database (release 26·0) with cystic fibrosis transmembrane conductance regulator (PIR code A30300)

PIR code	Optimal PAM-250 alignment			Optimal PAM-250 score (bits)	Optimal PAM-120 score (bits)
A30300	438 IPVLKDINFKIERGQLLAVAGSTGAGKTSLLMMIMGELEPSEGKI 482				
S05328	18 VSKDINLEIQDGEFVVFVGPSGCGKSTLLRMIAGLETVTSGDL		60	28.3	40.0
BVECOA	11 THNLKHINLVIPRDKLIVVTGLSGSGKSSL		40	24.7	35.0
A30300	1219 YIEGGNAILENISFSISPGQRVGLLGRTGSGKSTLLSAFLRLLNTEGEI 1267				
QRECFH	19 FRVPGRTLLHELSLTFPAGKVTGLIGHNGSGKSTLLRMLGR		59	29.3	35.0
QREBOT	31 DGDVIAVNDINTTLRAGETLGIVGESGSGKSQSRLRMGLLAINGRI		77	28.3	32.5
	Highest chance alignment score:			31.3	33.0
	PIR code of sequence involved:			A34416	A32916

A30300, Cystic fibrosis transmembrane conductance regulator; S05328, *Enterobacter aerogenes* inner membrane protein malK (Dahl *et al.*, 1988); BVECOA, *Escherichia coli* uvrA protein (Husain *et al.*, 1986); QRECFH, ferrichrome-iron transport protein fhuC (Coulton *et al.*, 1987); QREBOT, oligopeptide permease membrane protein oppD (Higgins *et al.*, 1985); A34416, fluke hydroxymethylglutaryl-CoA reductase (NADPH) (Rajkovic *et al.*, 1989); A32916, *Vibrio harveyi* acyl-protein synthetase (fragment) (Johnston *et al.*, 1989).

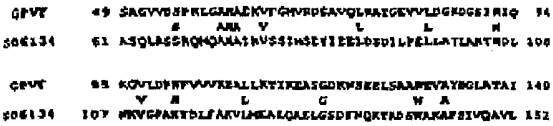


Figure 1. The PAM-250 maximal segment pair of broad bean leghemoglobin I (PIR code GPVF) and sea cucumber hemoglobin I (PIR code S06134). Identical residues are echoed on the central line. PAM-250 score: 25·3 bits; length: 92 residues.

sequence (1480 residues), and partly to its composition, which renders the parameter λ slightly smaller than in the previous examples.

(d) *Globins*

It is possible to find examples of long alignments representing distant relationships that are better distinguished by the PAM-250 than by the PAM-120 matrix. In practice such examples are rare, for some of the reasons discussed above. The globins are one superfamily in which sequence divergence has been relatively uniform over the length of entire proteins. As a result, some sequence relationships within this superfamily become apparent only with scoring systems tailored for long but very weak alignments.

For example, searching the PIR database with broad bean leghemoglobin I (PIR code GPVF; Richardson *et al.*, 1975), the alignment with sea cucumber hemoglobin I (PIR code S06134; Suzuki, 1989), shown in Figure 1, is found having a PAM-250 score of 25·3 bits. This is almost as high as the score of the best chance MSP (26·7 bits), which involves *Salmonella typhimurium* cystathionine β-lyase (PIR code JV0020; Park & Stauffer, 1989). The alignment is 92 residue pairs long; only 14 of these pairs involve identical amino acid residues, and they are spread fairly evenly along the alignment. This particular similarity is totally obscured when PAM-120 scores are used. The best region of the alignment shown then involves residues 100 to 133 of the leghemoglobin sequence and has a score of only 13 bits, while the best chance PAM-120 alignment, involving mouse hepatitis virus E1 membrane glycoprotein (PIR code VGIHE1; Armstrong *et al.*, 1984), scores 27·5 bits. Nevertheless, as in the previous examples, a number of relationships are distinguished by the PAM-120 matrix but missed by the PAM-250.

9. Conclusion

This paper has analyzed the properties of amino acid substitution matrices in the context of local alignments lacking gaps. This is exactly the sort of alignment sought by the recently developed BLAST database search programs (Altschul *et al.*, 1990; Altschul & Lipman, 1990). We have concluded that for protein databases of typical current size (about 1×10^7 residues), the most broadly sensitive substitution matrix should be a log-odds matrix with relative entropy of about one bit, e.g. the PAM-120 matrix. In order to detect short but strong homologies or long but weak ones, this matrix can be complemented by the PAM-40 and PAM-250 matrices; additional matrices should be of only marginal utility. Of course, many database search methods, such as the FASTA programs (Lipman & Pearson, 1985; Pearson & Lipman, 1988), seek local alignments with gaps, and such measures are potentially more sensitive to distant homologies. Unfortunately, if gaps with associated scores are allowed, the specific quantitative discussion above is no longer correct. Nevertheless, the general thrust of the arguments should still apply, and theory and experiment suggest that analogous results will hold for local alignments with gaps (Smith *et al.*, 1985; Waterman *et al.*, 1987; Collins *et al.*, 1988).

There are, of course, many much more involved ways for assessing local alignment than those discussed here. Scores can be assigned to aligned di-residues or tri-residues; they can depend on alignment length (Altschul & Erikson, 1986); or they can be complex combinations of various scoring methods (Argos, 1987). Protein databases may also be searched with position-dependent scores or "profiles" constructed from multiple alignments (Taylor, 1986; Gribskov *et al.*, 1987; Patthy, 1987). In certain contexts such systems may well be more sensitive than the straightforward local scoring system considered here. Two advantages of simple additive scores are their amenability to powerful algorithmic methods (Altschul *et al.*, 1990) and to rigorous statistical analysis (Karlin & Altschul, 1990; Karlin *et al.*, 1990). Such analysis may also yield insight into the properties of more complicated scoring schemes.

The author thanks Drs David Lipman, Mark Boguski and Andrew McLachlan for helpful conversations and suggestions on the manuscript.

References

Altschul, S. F. & Erickson, B. W. (1986). A nonlinear measure of subalignment similarity and its significance levels. *Bull. Math. Biol.* **48**, 617–632.

Altschul, S. F. & Lipman, D. J. (1990). Protein database searches for multiple alignments. *Proc. Nat. Acad. Sci., U.S.A.* **87**, 5509–5513.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410.

Argos, P. (1987). A sensitive procedure to compare amino acid sequences. *J. Mol. Biol.* **193**, 385–396.

Armstrong, J., Niemann, H., Smeekens, S., Rottier, P. & Warren, G. (1984). Sequence and topology of a model intracellular membrane protein, E1 glycoprotein, from a coronavirus. *Nature (London)*, **308**, 751–752.

Arratia, R. & Waterman, M. S. (1989). The Erdus–Renyi strong law for pattern matching with a given proportion of mismatches. *Ann. Prob.* **17**, 1152–1169.

Arratia, R., Gordon, L. & Waterman, M. S. (1986). An extreme value theory for sequence matching. *Ann. Stat.* **14**, 971–993.

Arratia, R., Morris, P. & Waterman, M. S. (1988). Stochastic scrabble: large deviations for sequences with scores. *J. Appl. Prob.* **25**, 106–119.

Boguski, M. S. & States, D. J. (1990). Molecular sequence databases and their uses. In *Protein Engineering: A Practical Approach* (Rees, A. R., Wetzel, R. & Sternberg, M. J. E., eds), chap. 5. IRL Press, Oxford.

Brooks, D. E., Means, A. R., Wright, E. J., Singh, S. P. & Tiver, K. K. (1986). Molecular cloning of the cDNA for two major androgen-dependent secretory proteins of 18·5 kilodaltons synthesized by the rat epididymis. *J. Biol. Chem.* **261**, 4956–4961.

Collins, J. F., Coulson, A. F. W. & Lyall, A. (1988). The significance of protein sequence similarities. *Comput. Appl. Biosci.* **4**, 67–71.

Coulton, J. W., Mason, P. & Allatt, D. D. (1987). *fhuC* and *fhuD* genes for iron(III)-ferrichrome transport into *Escherichia coli* K-12. *J. Bacteriol.* **169**, 3844–3849.

Cowan, S. W., Newcomer, M. E. & Jones, T. A. (1990). Crystallographic refinement of human serum retinol binding protein at 2 Å resolution. *Proteins*, **8**, 44–61.

Dahl, M. K., Franzoi, E., Saurin, W., Boos, W., Manson, M. D. & Hofnung, M. (1989). Comparison of sequences from the malB regions of *Salmonella typhimurium* and *Enterobacter aerogenes* with *Escherichia coli* K12: a potential new regulatory site in the interoperonic region. *Mol. Gen. Genet.* **218**, 199–207.

Dayhoff, M. O., Schwartz, R. M. & Orcutt, B. C. (1978). A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure* (Dayhoff, M. O., ed.), vol. 5, suppl. 3, pp. 345–352. Nat. Biomed. Res. Found., Washington, DC.

Dembo, A. & Karlin, S. (1991). Strong limit laws of empirical functionals for large exceedences of partial sums of I.I.D. variables. *Ann. Prob.* In the press.

Drayna, D. T., McLean, J. W., Wion, K. L., Trent, J. M., Drabkin, H. A. & Lawn, R. M. (1987). Human apolipoprotein D gene: gene sequence, chromosome localization, and homology to the α_{2u}-globulin superfamily. *DNA*, **6**, 199–204.

Feng, D. F., Johnson, M. S. & Doolittle, R. F. (1985). Aligning amino acid sequences: comparison of commonly used methods. *J. Mol. Evol.* **21**, 112–125.

Goad, W. B. & Kanehisa, M. I. (1982). Pattern recognition in nucleic acid sequences. I. A general method for finding local homologies and symmetries. *Nucl. Acids Res.* **10**, 247–263.

Gribskov, M., McLachlan, A. D. & Eisenberg, D. (1987). Profile analysis: detection of distantly related proteins. *Proc. Nat. Acad. Sci., U.S.A.* **84**, 4355–4358.

Higgins, C. F., Hiles, I. D., Whalley, K. & Jamieson, D. J. (1985). Nucleotide binding by membrane components of bacterial periplasmic binding protein-dependent transport systems. *EMBO J.* **4**, 1033–1039.

Higgins, C. F., Hiles, I. D., Salmond, G. P., Gill, D. R., Downie, J. A., Evans, I. J., Holland, I. B., Gray, L., Buckel, S. D., Bell, A. W. & Hermodson, M. A. (1986). A family of related ATP-binding subunits coupled to many distinct biological processes in bacteria. *Nature (London)*, **323**, 448–450.

Holmquist, R., Goodman, M., Conroy, T. & Czelusniak, J. (1983). The spatial distribution of fixed mutations within genes coding for proteins. *J. Mol. Evol.* **19**, 437–448.

Hussain, J., Van Houten, B., Thomas, D. C. & Sancar, A. (1986). Sequences of *Escherichia coli uvrA* gene and protein reveal two potential ATP binding sites. *J. Biol. Chem.* **261**, 4895–4901.

Ishioka, N., Takahashi, N. & Putnam, F. W. (1986). Amino acid sequence of human plasma α 1B-glycoprotein: homology to the immunoglobulin supergene family. *Proc. Nat. Acad. Sci., U.S.A.* **83**, 2363–2367.

Johnston, T. C., Hruska, K. S. & Adams, L. F. (1989). The nucleotide sequence of the *luxE* gene of *Vibrio harveyi* and a comparison of the amino acid sequences of the acyl-protein synthetases from *V. harveyi* and *V. fischeri*. *Biochem. Biophys. Res. Commun.* **163**, 93–101.

Karlin, S. & Altschul, S. F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Nat. Acad. Sci., U.S.A.* **87**, 2264–2268.

Karlin, S., Dembo, A. & Kawabata, T. (1990). Statistical composition of high-scoring segments from molecular sequences. *Ann. Stat.* **18**, 571–581.

Kaumeyer, J. F., Polazzi, J. O. & Kotick, M. P. (1986). The mRNA for a proteinase inhibitor related to the H1-30 domain of inter-α-trypsin inhibitor also encodes α-1-microglobulin (protein HC). *Nucl. Acids Res.* **14**, 7839–7850.

Lipman, D. J. & Pearson, W. R. (1985). Rapid and sensitive protein similarity searches. *Science*, **227**, 1435–1441.

McLachlan, A. D. (1971). Tests for comparing related amino acid sequences. Cytochrome c and cytochrome c_{551}. *J. Mol. Biol.* **61**, 409–424.

Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.* **48**, 443–453.

Osorio-Keese, M. E., Keese, P. & Gibbs, A. (1989). Nucleotide sequence of the genome of eggplant mosaic tymovirus. *Virology*, **172**, 547–554.

Park, Y. M. & Stauffer, G. V. (1989). DNA sequence of the *metC* gene and its flanking regions from *Salmonella typhimurium* LT2 and homology with the corresponding sequence of *Escherichia coli*. *Mol. Gen. Genet.* **216**, 164–169.

Patthy, L. (1987) Detecting homology of distantly related proteins with consensus sequences. *J. Mol. Biol.* **198**, 567–577.

Pearson, W. R. & Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proc. Nat. Acad. Sci., U.S.A.* **85**, 2444–2448.

Pech, M. & Zachau, H. G. (1984). Immunoglobulin genes of different subgroups are interdigitated within the VK locus. *Nucl. Acids Res.* **12**, 9229–9236.

Peitsch, M. C. & Boguski, M. S. (1990). Is apolipoprotein D a mammalian bilin-binding protein? *New Biologist*, **2**, 197–206.

Qiu, F., Ray, P., Brown, K., Barker, P. E., Jhanwar, S., Ruddle, F. H. & Besmer, P. (1988). Primary structure of c-kit: relationship with the CSF-1/PDGF receptor kinase family–oncogenic activation of v-kit involves deletion of extracellular domain and C terminus. *EMBO J.* **7**, 1003–1011.

Rajkovic, A., Simonsen, J. N., Davis, R. E. & Rottman, F. M. (1989). Molecular cloning and sequence analysis of 3-hydroxy-3-methylglutaryl-coenzyme A reductase from the human parasite *Schistosoma*

mammoni. Proc. Nat. Acad. Sci., U.S.A. **86**, 8217–8221.

Rao, J. K. M. (1987). New scoring matrix for amino acid residue exchanges based on residue characteristic physical parameters. *Int. J. Pept. Protein Res.* **29**, 276–281.

Richardson, M., Dilworth, M. J. & Scawen, M. D. (1975). The amino acid sequence of leghaemoglobin I from root nodules of broad bean (*Vicia faba* L.). *FEBS Letters*, **51**, 33–37.

Riordan, J. R., Rommens, J. M., Kerem, B. S., Alon, N., Rozmahel, R., Grzelczak, Z., Zielenski, J., Lok, S., Plavsic, N., Chou, J. L., Drumm, M. L., Iannuzzi, M. C., Collins, F. S. & Tsui, L. C. (1989). Identification of the cystic fibrosis gene: cloning and characterization of complementary DNA. *Science*, **245**, 1066–1073.

Risler, J. L., Delorme, M. O., Delacroix, H. & Henaut, A. (1988). Amino acid substitutions in structurally related proteins. A pattern recognition approach. Determination of a new and efficient scoring matrix. *J. Mol. Biol.* **204**, 1019–1029.

Sankoff, D. & Kruskal, J. B. (1983). *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading, MA.

Schwartz, R. M. & Dayhoff, M. O. (1978). Matrices for detecting distant relationships. In *Atlas of Protein Sequence and Structure* (Dayhoff, M. O., ed.), vol. 5, suppl. 3, pp. 353–358. Nat. Biomed. Res. Found., Washington, DC.

Sellers, P. H. (1974). On the theory and computation of evolutionary distances. *SIAM J. Appl. Math.* **26**, 787–793.

Sellers, P. H. (1984). Pattern recognition in genetic sequences by mismatch density. *Bull. Math. Biol.* **46**, 501–514.

Simmons, D. & Seed, B. (1988). The Fcγ receptor of natural killer cells is a phospholipid-linked membrane protein. *Nature (London)*, **333**, 568–570.

Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. *J. Mol. Biol.* **147**, 195–197.

Smith, T. F., Waterman, M. S. & Burks, C. (1985). The statistical distribution of nucleic acid similarities. *Nucl. Acids Res.* **13**, 645–656.

Stormo, G. D. & Hartzell, G. W., III (1989). Identifying protein-binding sites from unaligned DNA fragments. *Proc. Nat. Acad. Sci., U.S.A.* **86**, 1183–1187.

Suzuki, T. (1989). Amino acid sequence of a major globin from the sea cucumber *Paracaudina chilensis*. *Biochim. Biophys. Acta*, **998**, 292–296.

Taylor, W. R. (1986). Identification of protein sequence homology by consensus template alignment. *J. Mol. Biol.* **188**, 233–258.

Urade, Y., Nagata, A., Suzuki, Y. & Hayaishi, O. (1989). Primary structure of rat brain prostaglandin D synthetase deduced from cDNA sequence. *J. Biol. Chem.* **264**, 1041–1045.

Uzzell, T. & Corbin, K. W. (1971). Fitting discrete probability distributions to evolutionary events. *Science*, **172**, 1089–1096.

Van de Weghe, A., Coppieters, W., Bauw, G., Vanderkerckhove, J. & Bouquet, Y. (1988). The homology between the serum proteins PO2 in pig, Xk in horse and α_1B-glycoprotein in human. *Comp. Biochem. Physiol.* **90B**, 751–756.

Waterman, M. S., Gordon, L. & Arratia, R. (1987). Phase transitions in sequence matches and nucleic acid structure. *Proc. Nat. Acad. Sci., U.S.A.* **84**, 1239–1243.

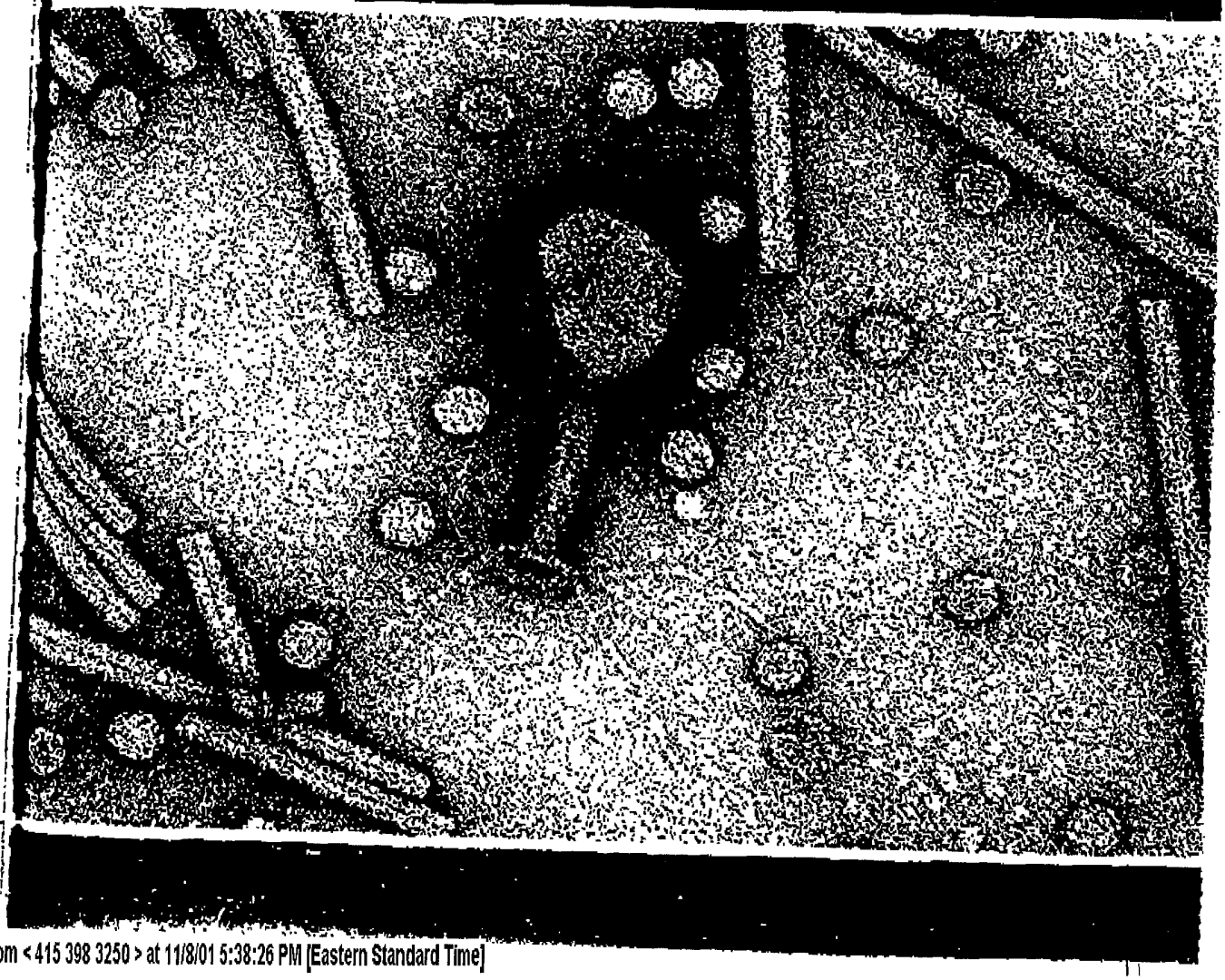
Wilbur, W. J. (1985). On the PAM matrix model of protein evolution. *Mol. Biol. Evol.* **2**, 434–447.

Zalacain, M., Gonzalez, A., Guerrero, M. C., Mattaliano, R. J., Malpartida, F. & Jimenez, A. (1986). Nucleotide sequence of the hygromycin B phosphotransferase gene from *Streptomyces hygroscopicus*. *Nucl. Acids Res.* **14**, 1565–1581.

Edited by F. E. Cohen

CORE

Journal of MOLECULAR BIOLOGY



dealing
vels of
/. X-ray
r. imag-
ielding

uctural
tments.
ire and
lar and
uctural

s struc-
mmunc-
ination
ncour-
nts that
e also
utions
onents
usually
ogress

: $444.00
. $535.00

Journal of Molecular Biology

Volume 215, Number 3

Contents